AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions of claims in the application:

Listing of Claims:

1. (Currently Amended) A full-text search computer implemented system comprising:

a plug-in component that defines a relevant score algorithm and a full-text index schema, wherein the relevance algorithm facilitates ranking matching documents and providing a list of documents in order of their relevance; and

a search component to receive and utilize the plug-in component to query data from a data store, populate an index in accordance with the provided index schema and utilize the index to generate a list of matching documents in order of their relevance as specified by the relevant score algorithm.

2. (Cancelled)

3. (Original) The system of claim 2, wherein the plug-in component specifies how the schema is to be populated.

4. (Original) The system of claim 1, wherein the search component is tightly integrated into a database management system.

5. (Original) The system of claim 4, wherein the database management system query execution engine executes received full-text queries and database queries.

6. (Original) The system of claim 5, wherein the full-text queries are optimized by the database management system optimization component.

7. (Original) The system of claim 1, wherein the search component comprises an index system that creates an index in accordance with one or more consumer plug-in components and a query processing system that provides a mechanism for retrieving relevant documents utilizing the index.

8. (Original) The system of claim 7, wherein the index is a compressed nested data structure.

9. (Previously Presented) A full-text indexing computer implemented system comprising:

a gatherer component to retrieve a document from a data store;

a producer pipeline component that parses the structure and text of the retrieved document in accordance with a plurality of third party developer specified components; and

a consumer pipeline component that receives data from the producer pipeline component and persists data to an inverted index, upon receipt of data, the consumer pipeline component consumes the data and takes action via executing a message through transacted message queues.

10. (Original) The system of claim 9, wherein the gather component and the consumer pipeline component reside within a database management system.

11. (Original) The system of claim 10, wherein the producer pipeline component is executed as an external daemon process managed by an external host controller component residing within the database management system.

12. (Original) The system of claim 9, wherein the producer pipeline component comprises a noise component to remove keywords that are diminutive in value as search criteria.

13. (Original) The system of claim 9, wherein the gatherer component retrieves documents from external databases.

14. (Currently Amended) A full text query computer implemented system tightly integrated with a database management system comprising:

a parser component that tokenizes received queries;

an execution plan generation system that generates an execution plan based on tokens received from the parser component and a ranking algorithm provided by a third party developer *via* a ranking plug-in component, wherein the ranking algorithm ranks documents from most to least relevant;

an execution engine component that utilizes the execution plan to search an index and produce a list of matching documents in order as specified by the ranking algorithm.

15. (Original) The system of claim 14, further comprising a user interface component to receive queries from users.

16. (Original) The system of claim 14, wherein execution plan generator component produces a query tree, appends scoring functions to leaves, and transforms the query tree into a data base query expression.

17. (Original) The system of claim 16, further comprising an optimizer component that optimizes the database query structure based on information concerning the index to be searched.

18. (Original) The system of claim 17, further comprising an expander component that modifies keywords provided in the query structure.

19. (Original) The system of claim 18, wherein the expander component is executed as a separate daemon process from the execution engine.

20. (Original) The system of claim 19, wherein the expander component includes at least one of a stemmer component, a normalizer component, an inflection component, a thesaurus component, a custom expansion component, a homonym component, and a fuzzy component.

21. (Original) The system of claim 20, where the expander components and associated functionality are specified by a third party developer.

22. (Previously Presented) A computer implemented method of employing a customized full-text query comprising:

retrieving a full-text indexing schema and ranking algorithm from a plug-in component provided by a third party developer;

populating an index in accordance with the provided indexing schema;

receiving a query;

generating a list of documents utilizing the index; and

displaying the list of documents [[by]] ranked in accordance with their relevance by the ranking algorithm, wherein the list of documents are displayed in order from most to least relevant.

23. (Cancelled)

24. (Previously Presented) The method of claim 22, wherein the results are generated by a database management system query processor.

25. (Cancelled)

26. (Currently Amended) A customized indexing computer implemented methodology comprising:

retrieving a document from a data source;

removing document formatting data and emitting text chunks;

parsing the text chucks into keywords;

persisting the keywords to an index, the index schema being defined by a third party developer;

employing a ranking algorithm to facilitate ranking matching documents<u>, wherein the ranking algorithm ranks documents from most to least relevant</u>; and displaying a list of documents by rank in accordance with the ranking algorithm~~, wherein the list of documents are displayed in order from most to least relevant~~.

27. (Original) The method of claim 26, further comprising normalizing keywords.

28. (Original) The method of claim 26, further comprising removing keywords of diminutive value as search criteria.

29. (Original) The method of claim 26, further comprising identifying the language of each text chunk and generating an id indicative thereof.

30. (Original) The method of claim 26, further comprising determining and noting the position of each keyword within a respective document.

31. (Cancelled).

32. (Currently Amended) A full text search computer implemented methodology comprising:

receiving a search request;

generating a query expression in response to the search request which includes a third party developer specified ranking algorithm for determining the relevance of result documents, wherein the ranking algorithm ranks documents from most to least relevant; and

displaying a list of documents by rank in accordance with the ranking algorithm~~, wherein the list of documents are displayed in order from most to least relevant~~.

33. (Original) The method of claim 32, further comprising utilizing a database management system to optimize the query expression prior to execution.

34. (Original) The method of claim 33, further comprising employing a database execution engine to execute full text queries.

35. (Original) The method of claim 32, further comprising modifying the query expression to include, remove, add, or modify keyword terms.

36. (Original) The method of claim 35, wherein the query expression is modified by one or more components specified by a third party developer.

37. (Currently Amended) The method of claim 36, wherein the <u>third party</u> developer specified components are executed as separate daemon processes managed by an external host controller component from within a database management system.

38. (Original) The method of claim 35, wherein the query expression is modified once at compile time and again at runtime.

39. (Cancelled)

40. (Cancelled)